

## Freeform Search

---

<b>Database:</b>	US Pre-Grant Publication Full-Text Database <b>US Patents Full-Text Database</b> US OCR Full-Text Database EPO Abstracts Database JPO Abstracts Database Derwent World Patents Index IBM Technical Disclosure Bulletins
<b>Term:</b>	(host\$ same (virtual adj1 server\$))
<b>Display:</b>	<input type="text" value="10"/> Documents in <b>Display Format:</b> <input type="text" value="KWIC"/> Starting with Number <input type="text" value="2"/>
<b>Generate:</b> <input type="radio"/> Hit List <input checked="" type="radio"/> Hit Count <input type="radio"/> Side by Side <input type="radio"/> Image	

---

---

### Search History

---

DATE: Wednesday, December 24, 2003   [Printable Copy](#)   [Create Case](#)

#### Set Name Query

side by side

*DB=USPT; PLUR=YES; OP=ADJ*

#### Hit Count Set Name

result set

<u>L12</u> (host\$ same (virtual adj1 server\$))	27	<u>L12</u>
<u>L11</u> 6286047[uref]	4	<u>L11</u>
<u>L10</u> 6502135[uref]	1	<u>L10</u>
<u>L9</u> 6286047[uref]	4	<u>L9</u>
<u>L8</u> 6647422[uref]	0	<u>L8</u>
<u>L7</u> (virtual adj1 servers) and L1	66	<u>L7</u>
<u>L6</u> (virtual adj1 servers).ab.	8	<u>L6</u>
<u>L5</u> (virtual adj1 servers)	116	<u>L5</u>
<u>L4</u> (plurality with (virtual adj1 servers))	9	<u>L4</u>
<u>L3</u> (host same (plurality with (virtual adj1 servers)))	2	<u>L3</u>
<u>L2</u> L1 and (host same (plurality with (virtual adj1 servers)))	2	<u>L2</u>
<u>L1</u> 709/\$.cccls.	17513	<u>L1</u>

END OF SEARCH HISTORY

## Refine Search

### Search Results -

Term	Documents
VIRTUAL	59527
VIRTUALS	3
SERVERS	19424
SERVER	41281
(1 AND (VIRTUAL ADJ1 SERVERS)).USPT.	66
((VIRTUAL ADJ1 SERVERS) AND L1).USPT.	66

Database:

US Pre-Grant Publication Full-Text Database  
 US Patents Full-Text Database  
 US OCR Full-Text Database  
 EPO Abstracts Database  
 JPO Abstracts Database  
 Derwent World Patents Index  
 IBM Technical Disclosure Bulletins

Search:

L7

Refine Search

Recall Text

Clear

Interrupt

### Search History

DATE: Wednesday, December 24, 2003    [Printable Copy](#)    [Create Case](#)

**Set Name Query**

side by side

DB=USPT; PLUR=YES; OP=ADJ

L7 (virtual adj1 servers) and L1

L6 (virtual adj1 servers).ab.

L5 (virtual adj1 servers)

L4 (plurality with (virtual adj1 servers))

L3 (host same (plurality with (virtual adj1 servers)))

L2 L1 and (host same (plurality with (virtual adj1 servers)))

L1 709/\$.ccls.

**Hit Count Set Name**

result set

66 L7

8 L6

116 L5

9 L4

2 L3

2 L2

17513 L1

END OF SEARCH HISTORY

## Refine Search

### Search Results -

Term	Documents
VIRTUAL	59527
VIRTUALS	3
HOST\$	0
HOST	133228
HOSTA	67
HOSTAAACS3N	1
HOSTAAPEAM	1
HOSTABLE	4
HOSTABUAR	1
HOSTAC	2
HOSTACAIN	1
((HOST\$ SAME (VIRTUAL ADJ1 SERVER\$)) ).USPT.	27

There are more results than shown above. [Click here to view the entire set.](#)

Database:

US Pre-Grant Publication Full-Text Database  
 US Patents Full-Text Database  
 US OCR Full-Text Database  
 EPO Abstracts Database  
 JPO Abstracts Database  
 Derwent World Patents Index  
 IBM Technical Disclosure Bulletins

Search:

L12





### Search History

DATE: Wednesday, December 24, 2003    [Printable Copy](#)    [Create Case](#)

#### Set Name Query

side by side

DB=USPT; PLUR=YES; OP=ADJ

L12    (host\$ same (virtual adj1 server\$))

#### Hit Count Set Name

result set

27    L12

<u>L11</u>	6286047[uref]	4	<u>L11</u>
<u>L10</u>	6502135[uref]	1	<u>L10</u>
<u>L9</u>	6286047[uref]	4	<u>L9</u>
<u>L8</u>	6647422[uref]	0	<u>L8</u>
<u>L7</u>	(virtual adj1 servers) and L1	66	<u>L7</u>
<u>L6</u>	(virtual adj1 servers).ab.	8	<u>L6</u>
<u>L5</u>	(virtual adj1 servers)	116	<u>L5</u>
<u>L4</u>	(plurality with (virtual adj1 servers))	9	<u>L4</u>
<u>L3</u>	(host same (plurality with (virtual adj1 servers)))	2	<u>L3</u>
<u>L2</u>	L1 and (host same (plurality with (virtual adj1 servers)))	2	<u>L2</u>
<u>L1</u>	709/\$.ccls.	17513	<u>L1</u>

END OF SEARCH HISTORY

## Refine Search

Your wildcard search against 10000 terms has yielded the results below.

***Your result set for the last L# is incomplete.***

The probable cause is use of unlimited truncation. Revise your search strategy to use limited truncation.

### Search Results -

Term	Documents
VIRTUAL	59511
VIRTUALS	3
WAN\$	0
WAN	13569
WANA	16
WANAA	1
WANAAO	1
WANABE	5
WANABE-OUTDOORS	1
WANABE-OUTDOORS-INC	3
WANABISHI	1
(L1 AND (VIRTUAL ADJ1 (WAN\$ OR LAN\$)).AB.).USPT.	12

There are more results than shown above. Click here to view the entire set.

Database:

US Pre-Grant Publication Full-Text Database  
US Patents Full-Text Database  
US OCR Full-Text Database  
EPO Abstracts Database  
JPO Abstracts Database  
Derwent World Patents Index  
IBM Technical Disclosure Bulletins

Search:

L5

Refine Search

Recall Text

Clear

Interrupt

### Search History

DATE: Wednesday, December 24, 2003   [Printable Copy](#)   [Create Case](#)

**Set Name Query**  
side by side

**Hit Count Set Name**  
result set

h   e b   b   c g b   e e ch

*DB=USPT; PLUR=YES; OP=ADJ*

<u>L5</u>	L1 and (virtual adj1 (wan\$ or lan\$)).ab.	12	<u>L5</u>
<u>L4</u>	L1 and (remote\$ with (virtual adj1 (wan\$ or lan\$)))	3	<u>L4</u>
<u>L3</u>	L1 and (remote\$ with (virtual adj1 (wan\$ or lan\$)))	3	<u>L3</u>
<u>L2</u>	L1 and (remote\$ with (virtual adj1 (wan\$ or lan\$)))	3	<u>L2</u>
<u>L1</u>	709/\$.ccls.	17513	<u>L1</u>

END OF SEARCH HISTORY

**WEST**☐

Generate Collection

L5: Entry 1 of 2

File: USPT


Aug 22, 2000

DOCUMENT-IDENTIFIER: US 6108701 A

TITLE: Soft switch extension for internet protocol applications

Detailed Description Text (2):

With reference now to the figures and in particular with reference to FIG. 2A, a simplified schematic of the preferred multiple server embodiment of the present invention is depicted. In this embodiment, a plurality of servers 200, 202 are assigned the same IP address on a local area network ("LAN") 204. A router 206 is coupled to the LAN and connects the servers 200, 202 via an Intranet or Internet 208 to a client computer 210. Also coupled to the LAN is a DNS Server 212 that is used for "name to IP address" translation. The plurality of servers 200, 202 thus appear as one logical host (i.e. a virtual server) 214 to the client 210 and also to the DNS Server 212, because the servers 200, 202 share the same IP address and host name.

Set Name Query  
side by sideHit Count Set Name  
result set*DB=USPT; PLUR=YES; OP=ADJ*

<u>L5</u>	L1 and ((virtual adj1 server\$) with host\$ with figure\$)	0	<u>L5</u>
<u>L4</u>	L1 and ((virtual adj1 server\$) with host\$)	15	<u>L4</u>
<u>L3</u>	L1 and ((virtual adj1 server\$) same host\$)	16	<u>L3</u>
<u>L2</u>	L1 and (virtual adj1 server\$.ab.	5	<u>L2</u>
<u>L1</u>	((709/\$)!.CCLS.)	16227	<u>L1</u>

END OF SEARCH HISTORY

L3 #6 6, 247, 057



**WEST**

Generate Collection

Print

**Search Results - Record(s) 1 through 10 of 16 returned.**☐ 1. Document ID: US 6553413 B1

L3: Entry 1 of 16

File: USPT

Apr 22, 2003

DOCUMENT-IDENTIFIER: US 6553413 B1

TITLE: Content delivery network using edge-of-network servers for providing content delivery to a set of participating content providers

Brief Summary Text (26):

According to the present invention, load balancing across the set of hosting servers is achieved in part through a novel technique for distributing the embedded object requests. In particular, each embedded object URL is preferably modified by prepending a virtual server hostname into the URL. More generally, the virtual server hostname is inserted into the URL. Preferably, the virtual server hostname includes a value (sometimes referred to as a serial number) generated by applying a given hash function to the URL or by encoding given information about the object into the value. This function serves to randomly distribute the embedded objects over a given set of virtual server hostnames. In addition, a given fingerprint value for the embedded object is generated by applying a given hash function to the embedded object itself. This given value serves as a fingerprint that identifies whether the embedded object has been modified. Preferably, the functions used to generate the values (i.e., for the virtual server hostname and the fingerprint) are applied to a given Web page in an off-line process. Thus, when an HTTP request for the page is received, the base HTML document is served by the Web site and some portion of the page's embedded objects are served from the hosting servers near (although not necessarily the closest) to the client machine that initiated the request.

Detailed Description Text (11):

The routine begins at step 50 by determining whether all of the embedded objects in a given page have been processed. If so, the routine ends. If not, however, the routine gets the next embedded object at step 52. At step 54, a virtual server hostname is prepended into the URL for the given embedded object. The virtual server hostname includes a value (e.g., a number) that is generated, for example, by applying a given hash function to the URL. As is well-known, a hash function takes arbitrary length bit strings as inputs and produces fixed length bit strings (hash values) as outputs. Such functions satisfy two conditions: (1) it is infeasible to find two different inputs that produce the same hash value, and (2) given an input and its hash value, it is infeasible to find a different input with the same hash value. In step 54, the URL for the embedded object is hashed into a value xx,xxx that is then included in the virtual server hostname. This step randomly distributes the object to a given virtual server hostname.

Detailed Description Text (12):

The present invention is not limited to generating the virtual server hostname by applying a hash function as described above. As an alternative and preferred embodiment, a virtual server hostname is generated as follows. Consider the representative hostname al234.g.akamaitech.net. The 1234 value, sometimes referred to as a serial number, preferably includes information about the object such as its size (big or small), its anticipated popularity, the date on which the object was created, the identity of the Web site, the type of object (e.g., movie or static picture), and perhaps some random bits generated by a given random function. Of course, it is not required that any given serial number encode all of such

information or even a significant number of such components. Indeed, in the simplest case, the serial number may be a simple integer. In any event, the information is encoded into a serial number in any convenient manner. Thus, for example, a first bit is used to denote size, a second bit is used to denote popularity, a set of additional bits is used to denote the date, and so forth. As noted above in the hashing example, the serial number is also used for load balancing and for directing certain types of traffic to certain types of servers. Typically, most URLs on the same page have the same serial number to minimize the number of distinguished name (DN) accesses needed per page. This requirement is less important for larger objects.

Detailed Description Text (13):

Thus, according to the present invention, a virtual server hostname is prepended into the URL for a given-embedded object, and this hostname includes a value (or serial number) that is generated by applying a given function to the URL or object. That function may be a hash function, an encoding function, or the like.

Current US Original Classification (1):

709/219

Current US Cross Reference Classification (1):

709/200

Current US Cross Reference Classification (2):

709/217

Current US Cross Reference Classification (3):

709/218

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KNAC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	-----------	-------

☐ 2. Document ID: US 6553409 B1

L3: Entry 2 of 16

File: USPT

Apr 22, 2003

*same as 7*

DOCUMENT-IDENTIFIER: US 6553409 B1

TITLE: Background cache synchronization

Detailed Description Text (12):

Turning to the drawings, FIG. 2 shows an architecture for caching content into which the present invention may be incorporated. In FIG. 2, a network application 60 (or operating system) such as including a browser, is loaded in a client machine (e.g., the personal computer system 20), and communicates via APIs 61 and a network interface 62 with a server (e.g., the remote computer 49) in order to download content 64 therefrom. Communication between the client 20 and the server 49 may take place using one of several well-known network protocols, such as hypertext transfer protocol (HTTP), file transfer protocol (FTP), Common Internet File System (CIFS) protocol, or Gopher, although for purposes of simplicity, the invention will be primarily described with respect to HTTP. Content available through these protocols may also be downloaded from the server to the client by alternative means, such as a multicast protocol or CD-ROM installation, for example. As used herein, "server" or "network server" includes any machine or combination of machines having content thereon. Network servers may thus include HTTP "web sites," including those having sites with different names (which may be regarded as different virtual servers even if they are hosted on the same physical machine). Note that a web site may be distributed over many virtual servers, which in turn may be distributed over many physical machines.

Current US Original Classification (1):

709/213Current US Cross Reference Classification (1):709/214

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KMC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	-----	-----------	-------

☐ 3. Document ID: US 6336138 B1

L3: Entry 3 of 16

File: USPT

Jan 1, 2002

DOCUMENT-IDENTIFIER: US 6336138 B1

TITLE: Template-driven approach for generating models on network services

Detailed Description Text (82):

The key difference between the approach of using network probes and the approach of using special-purpose discovery agents is that unlike the network probes, the discovery agents do not snoop on packet transmissions. Instead, the discovery agents use a number of operating systems and application-specific mechanisms to discover inter-service dependencies. These mechanisms include (1) processing service configuration information and (2) application-dependent monitoring tools. Referring first to the processing service configuration information, application servers determine their dependencies on other services from one or more configuration files. By processing the content of the configuration files, discovery agents can discover inter-service dependencies. An example of this is the processing of the web server's configuration file to discover whether it has a dependency on an NFS service. While processing the web server's configuration file, the discovery agent can also determine if the same application is being used to host multiple "virtual" servers (which is commonly used by ISPs to host web sites on behalf of their business customers). Typically, web server configuration files are specific to the type of server executed on the web server in use. The server type determination performed during external discovery (i.e., the first phase of discovery) is used for deciding the location and format of the configuration files.

Current US Original Classification (1):709/223Current US Cross Reference Classification (3):709/224Current US Cross Reference Classification (4):709/226

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KMC	Draw Desc	Image
-----	-----------	-------

☐ 4. Document ID: US 6327622 B1

L3: Entry 4 of 16

File: USPT

Dec 4, 2001

DOCUMENT-IDENTIFIER: US 6327622 B1

TITLE: Load balancing in a network environment

Brief Summary Text (7):

In another method of load balancing, specialized hardware is employed to store

information concerning the servers hosting instances of a replicated service. In particular, according to this method information is stored on a computer system other than the system that initially receives clients' requests. The stored information helps identify the server having the smallest load (e.g., fewest client requests). Based on that information, a user's request is routed to the least-loaded server. In a web-browsing environment, for example, when a user's service access request (e.g., a connection request to a particular Uniform Resource Locator (URL) or virtual server name) is received by a server offering Domain Name Services (DNS), the DNS server queries or passes the request to the specialized hardware. Based on the stored information, the user's request is then forwarded to the least-loaded server offering the requested service.

Current US Original Classification (1):

709/228

Current US Cross Reference Classification (1):

709/105

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWOC	Draw Desc	Image
------	-----------	-------

---

☐ 5. Document ID: US 6286047 B1

L3: Entry 5 of 16

File: USPT

Sep 4, 2001

DOCUMENT-IDENTIFIER: US 6286047 B1

TITLE: Method and system for automatic discovery of network services

Abstract Text (1):

A method for identifying services, service elements and dependencies among the services and service elements includes executing first and second phases of discovery. In the first phase, the services and service elements are detected, as well as a first set of dependencies. The second phase is based on results of the first phase and is focused upon detecting inter-service dependencies, i.e., conditions in which proper operation of one service relies upon at least one other service. Various techniques may be used in executing the first phase, including accessing information in a domain name service (DNS) of the network to identify dependencies, as well as services and service elements. Discovery within the first phase may also be based upon recognizing naming conventions. Regarding the second phase, one approach to discovering inter-service dependencies is to deploy discovery agents implemented in computer software to access content of configuration files of applications detected in the first phase. Discovery agents may also be used to monitor connections completed via specified service elements detected in the first phase, such that other inter-service dependencies are identified. As an alternative or additional approach, network probes may be deployed to access information of data packets transmitted between service elements detected in the first phase, with the accessed packet information being used to detect inter-service dependencies. When information of the DNS is accessed in the first phase, the information is used as a basis for determining at least some of (1) groups of service elements that are generally equivalent with respect to executing a particular service within the network, (2) hosts supporting virtual hosting, (3) hosts supporting virtual servers, and (4) name servers.

Brief Summary Text (20):

The recognition of the naming conventions used by the network provides evidence of any virtual hosts or virtual servers. For example, an ISP may use a single host machine to support multiple customer web sites. While each customer web site may be associated with a unique IP address, there will be a naming pattern that identifies the common host machine. Naming conventions may also be used to recognize associations between terminal servers of an ISP and POP sites.

Detailed Description Text (88):

The key difference between the approach of using network probes and the approach of using special-purpose discovery agents is that unlike the network probes, the discovery agents do not snoop on packet transmissions. Instead, the discovery agents use a number of operating systems and application-specific mechanisms to discover inter-service dependencies. These mechanisms include (1) processing service configuration information and (2) application-dependent monitoring tools. Referring first to the processing service configuration information, application servers determine their dependencies on other services from one or more configuration files. By processing the content of the configuration files, discovery agents can discover inter-service dependencies. An example of this is the processing of the web server's configuration file to discover whether it has a dependency on an NFS service. While processing the web server's configuration file, the discovery agent can also determine if the same application is being used to host multiple "virtual" servers (which is commonly used by ISPs to host web sites on behalf of their business customers). Typically, web server configuration files are specific to the type of server executed on the web server in use. The server type determination performed during the first phase of discovery is used for deciding the location and format of the configuration files.

Detailed Description Text (99):

An ISP system that hosts web sites for business customers poses several challenges for discovery. Typically, each web site of a business customer of the ISP has a unique name (e.g., www.customer-domain.com). The ISP is typically authoritative for the customer domain, i.e., one or more of the ISP's name servers advertise the customer's domain to the global Internet. There are three different models for web hosting in an ISP system: (1) dedicated hosts; (2) virtual hosts; and (3) virtual servers. In the dedicated hosts model, the web site of the customer may be supported on one or more dedicated hosts at the site of the ISP, in which case, there are one or more IP addresses associated with the customer's web site. On the other hand, the virtual hosts model is an approach in which multiple customer web sites are supported using the same host machine in the ISP system. In this case, there is a unique IP address associated with each customer's web site. Using capabilities built into the newer operating systems, the ISP can set up multiple virtual interfaces that map to one of the physical interfaces on the host machine. Each virtual interface is associated with an IP address, which in turn maps to one of the virtual hosted web sites. The web application server configuration file defines the root directory corresponding to each customer's web site. When it receives an HTTP request, the web server processes the IP address of the server, which is specified in the HTTP request header, to determine which root directory is used for servicing the request.

Detailed Description Text (100):

With regard to the virtual servers model, such servers are found when all of the customer web sites supported using a single host machine have an IP address that is common to the host machine. To map an incoming request to a virtual web site, the web server application executing on the host exploits recent modifications made to the HTTP protocol in version 1.1. Web browsers that are compatible with HTTP/1.1 specify the web site being accessed as part of the HTTP request. Web servers that are compatible with HTTP/1.1 process the web site name and the request to determine which of the various virtual servers the request is destined for and, therefore, which of the many configurations (root directory, access list, etc.) must be used to service the request. To support this approach, the ISP associates the virtual servers with the IP address of the host using canonical name (CNAME) records in the DNS database.

Detailed Description Text (103):

Once the customer domains that are supported by the ISP are determined, the discovery process executes the first and second phase discovery methodologies to discover the hosts and services in the different customer domains. In order to enable service models to be created for web hosting services, it is essential to discover the virtual hosts and the virtual servers. There are two possible approaches to executing the discovery of the virtual hosts. In a first approach, first phase discovery is implemented by interpreting application server responses. A

key observation guiding this approach is that in an ISP system, only web servers support virtual hosting. That is, the email (POP3, SMTP), News, and FTP application servers typically do not support virtual hosting. When the email, News, and FTP application servers are targeted with active tasks during the first phase discovery process, they return the name of the host machine from which they are executed as part of the response. Since the email, News, and FTP application servers are not aware of the existence of the virtual hosts, when the servers execute on a host that supports other virtual hosts, the servers return the name of the host machine (not the names of the virtual hosts) as part of their response. To discover the virtual hosts within this first approach, the discovery process determines all the host names that exist in the ISP system. The discovery process then targets each of the host names, attempting to connect to the email, News, or FTP application servers. In the event that a connection succeeds, the discovery process logs the name or names returned by the application servers as part of their response. The host name corresponds to a virtual host if its host name in the DNS database does not match the name returned by the email, News, or FTP application servers in response to active tests. For a virtual host, the name returned by the email, News, or FTP application servers represents the identity of the host machine that supports the virtual host.

Detailed Description Text (105):

The virtual servers must also be discovered. All IP addresses that have multiple host names associated with them in the DNS database are candidates for hosting virtual servers. However, this is not a sufficient condition for identifying virtual servers, since many times multiple host names are associated with the same host for naming convenience or for other administrative purposes. A more reliable method of identifying virtual servers and hosts that support them is to use discovery agents that can process the web application server configuration files.

Current US Original Classification (1):

709/224

Current US Cross Reference Classification (5):

709/202

Current US Cross Reference Classification (6):

709/217

Current US Cross Reference Classification (7):

709/226

CLAIMS:

7. The method of claim 1 wherein said step of executing said first phase includes accessing information of a domain name service (DNS) of said network, including identifying at least two of (1) internal and external name servers, (2) round-robin service groups of said network, and (3) virtual servers and virtual hosts of said network.

11. A method of identifying elements, services and dependencies among said elements and services comprising steps of:

accessing information of a domain name service (DNS) of a network; and

utilizing said information of said DNS as a basis for determining a plurality of:

(a) a group of service elements that are generally equivalent with respect to executing a particular service within said network;

(b) a host supporting virtual hosting;

(c) a host supporting virtual servers; and

(d) name servers that are authoritative for a domain.

19. The system of claim 16 wherein said first discovery tools include software configured to access a DNS of said network and to retrieve information indicative of at least two of (1) name servers, (2) round-robin service groups, and (3) virtual servers and virtual hosts.

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

RMIC	Draw Desc	Image
------	-----------	-------

☐ 6. Document ID: US 6247057 B1

L3: Entry 6 of 16

File: USPT

Jun 12, 2001

DOCUMENT-IDENTIFIER: US 6247057 B1

**\*\* See image for Certificate of Correction \*\***

TITLE: Network server supporting multiple instance of services to operate concurrently by having endpoint mapping subsystem for mapping virtual network names to virtual endpoint IDs

Brief Summary Text (9):

Implicit in the conventional processes is the assumption that each machine will only be running one instance of the service. However, current server technology allows administrators to run multiple instances of the same service on a single machine. For example, a database administrator might wish to run multiple instances of the SQL database service so that more physical memory can be used than can be addressed by a single address space. In this case, one instance of SQL covers one address space and another instance of SQL covers another address space. To the client, each SQL instances functions as its own service running on its own machine. In this manner, the physical host server can be said to support multiple "virtual services" on multiple "virtual servers".

Brief Summary Text (10):

As another example, it is not uncommon for a Web server to support thousands of domains on the same Web service. To the client, however, each domain functions as its own service as if running on its own HTTP (Hypertext Transfer Protocol) server on its own machine. Here again, one physical host server is effectively running multiple "virtual services" on multiple "virtual servers".

Detailed Description Text (7):

In the exemplary illustration, two instances 42(1) and 42(2) are concurrently executing on the host server 22. Each instance 42(1) and 42(2) presents itself to the clients 24(1) and 24(2) as a "virtual service". Even though the virtual services are instances of the same service type on the same machine, each virtual service appears to the client as its own service running on its own machine. Hence, the service 42 is said to support multiple "virtual services"; or said another way, the host server 22 is said to present multiple "virtual servers".

Current US Original Classification (1):

709/229

Current US Cross Reference Classification (1):

709/203

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

RMIC	Draw Desc	Image
------	-----------	-------

☐ 7. Document ID: US 6233606 B1

L3: Entry 7 of 16

File: USPT

May 15, 2001

*done as 2*

DOCUMENT-IDENTIFIER: US 6233606 B1  
TITLE: Automatic cache synchronization

Detailed Description Text (12):

Turning to the drawings, FIG. 2 shows a generalized conceptual model of the present invention wherein a network application 60 such as a browser in a client machine (e.g., the personal computer system 20) communicates via APIs 61 and a network interface 62 with a server (e.g., the remote computer 49) in order to download content 64 therefrom. Communication between the client 20 and the server 49 may take place using one of several well-known network protocols, such as hypertext transfer protocol (HTTP), file transfer protocol (FTP), Common Internet File System (CIFS) protocol, or Gopher, although for purposes of simplicity, the invention will be primarily described with respect to HTTP. Content available through these protocols may also be downloaded from the server to the client by alternative means, such as a multicast protocols or CD-ROM installation, for example. As used herein, "server" or "network server" includes any machine or combination of machines having content thereon. Network servers may thus include HTTP "web sites," including those having sites with different names (which may be regarded as different virtual servers even if they are hosted on the same physical machine). Note that a web site may be distributed over many virtual servers, which in turn may be distributed over many physical machines.

Current US Original Classification (1):  
709/213

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWIC	Draw Desc	Image
------	-----------	-------

☐ 8. Document ID: US 6205477 B1

L3: Entry 8 of 16

File: USPT

Mar 20, 2001

DOCUMENT-IDENTIFIER: US 6205477 B1  
TITLE: Apparatus and method for performing traffic redirection in a distributed system using a portion metric

Detailed Description Text (27):

According to another embodiment of the invention, the distributed director operates in HTTP redirect mode in accordance with the HTTP protocol. Referring now to FIG. 6, a method for operating the distributed director in HTTP redirect mode 84 is presented. The distributed director is configured at step 86. Configuration may include associating each host with a plurality of IP addresses corresponding to a plurality of web servers in the distributed system, assigning a portion metric to each server, assigning other metrics to each server, and specifying a tolerance range for application of each specified metric as described above in step 28. Next, at step 88, an HTTP connection is accepted from a client. Thus, the distributed director operates as though it were a web server. The distributed director has an IP address as well as a different IP address used by a web redirector. As a result, the client may communicate with the web redirector by connecting to the web redirector's IP address. The web redirector may then redirect connections through the use of virtual web servers. By way of example, a different host (and IP address) is associated with each set of virtual servers. The client is then connected to one of a set of virtual web servers associated with the distributed director at step 89. The client's IP address is then obtained at step 90 since it is later used to send



an HTTP redirect. A host name (e.g., www.cisco.com) associated with the IP address of the virtual web server connected to is then determined at step 92. This may be performed via a DNS server, as described above.

Detailed Description Text (28):

Once the client is connected to the virtual web server, it is desirable to obtain a set of IP addresses from which to select an IP address and send an HTTP code redirect. One problem which may occur when a set of IP addresses associated with the host name of the virtual server is obtained is that the IP address of the virtual web server connected to may ultimately be selected. Since it would be extremely undesirable to select the virtual web server that the client is connected to, a new host name different from the host name of the virtual server connected to may be constructed. At step 93, a new host name associated with the host name determined in step 92 is constructed. By way of example, a string may be appended to the host name of the virtual server that the client is connected to. A set of IP addresses of real web servers associated with the new host name are then obtained at step 94. By way of example, the real web servers may be associated with the new host name in a DNS server. In this manner, the IP address of the virtual web server connected to may be excluded, or filtered, from selection.

Current US Original Classification (1):

709/220

Current US Cross Reference Classification (1):

709/203

Current US Cross Reference Classification (2):

709/223

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWIC	Draw Desc	Image
------	-----------	-------

☐ 9. Document ID: US 6199107 B1

L3: Entry 9 of 16

File: USPT

Mar 6, 2001

DOCUMENT-IDENTIFIER: US 6199107 B1

TITLE: Partial file caching and read range resume system and method

Detailed Description Text (9):

FIG. 2 shows a generalized conceptual model of the present invention wherein a network application 60 such as a browser in a client machine (e.g., the personal computer system 20) communicates via APIs 61 and a network interface 62 with a server (e.g., the remote computer 49) in order to download content 64 therefrom. Communication between the client 20 and the server 49 preferably uses a well-known network protocol, such as hypertext transfer protocol (HTTP), and the network interface 62 preferably comprises the Wininet.dll application programming interface. As used herein, "server" or "network server" includes any machine or combination of machines having content thereon. Network servers may thus include HTTP "web sites," including those having sites with different names (which may be regarded as different virtual servers even if they are hosted on the same physical machine). Note that a web site may be distributed over many virtual servers, which in turn may be distributed over many physical machines.

Current US Original Classification (1):

709/219

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWIC	Draw Desc	Image
------	-----------	-------

☐ 10. Document ID: US 6189000 B1

L3: Entry 10 of 16

File: USPT

Feb 13, 2001

DOCUMENT-IDENTIFIER: US 6189000 B1

TITLE: System and method for accessing user properties from multiple storage mechanisms

Detailed Description Text (41):

For the Init routine, bszInstanceId is the virtual server identification and bszUserDN is the username. The OnStartPage function is called by a scripting host when first running a script. To properly initialize, the storage-mechanism interface receives the virtual server identification and the user's identification from a host in the scripting environment. Non-host applications can call Init and provide the virtual server identification and the user's identification.

Current US Cross Reference Classification (5):

709/203

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWIC	Draw Desc	Image
------	-----------	-------

[Generate Collection](#)[Print](#)

Term	Documents
VIRTUAL	56040
VIRTUALS	3
SERVERS	0
SERVER	36940
SERVERA	9
SERVERABILITY	1
SERVERABLE	11
SERVERABLY	2
SERVERACCEPT	1
SERVERACCESS	1
"SERVERACCESS.JAVA"	1
(L1 AND ((VIRTUAL ADJ1 SERVERS) SAME HOSTS)).USPT.	16

[There are more results than shown above. Click here to view the entire set.](#)**Display Format:**

KWIC

[Change Format](#)[Previous Page](#)[Next Page](#)

**WEST**[Generate Collection](#)[Print](#)**Search Results - Record(s) 11 through 16 of 16 returned.**☐ 11. Document ID: US 6182136 B1

L3: Entry 11 of 16

File: USPT

Jan 30, 2001

DOCUMENT-IDENTIFIER: US 6182136 B1

TITLE: Automated service elements discovery using core service specific discovery templates

Detailed Description Text (88):

The key difference between the approach of using network probes and the approach of using special-purpose discovery agents is that unlike the network probes, the discovery agents do not snoop on packet transmissions. Instead, the discovery agents use a number of operating systems and application-specific mechanisms to discover inter-service dependencies. These mechanisms include (1) processing service configuration information and (2) application-dependent monitoring tools. Referring first to the processing service configuration information, application servers determine their dependencies on other services from one or more configuration files. By processing the content of the configuration files, discovery agents can discover inter-service dependencies. An example of this is the processing of the web server's configuration file to discover whether it has a dependency on an NFS service. While processing the web server's configuration file, the discovery agent can also determine if the same application is being used to host multiple "virtual" servers (which is commonly used by ISPs to host web sites on behalf of their business customers). Typically, web server configuration files are specific to the type of server executed on the web server in use. The server type determination performed during external discovery (i.e., the first phase of discovery) is used for deciding the location and format of the configuration files.

Current US Original Classification (1):

709/224

Current US Cross Reference Classification (1):

709/202

Current US Cross Reference Classification (2):

709/218

Current US Cross Reference Classification (3):

709/220

Current US Cross Reference Classification (4):

709/223

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWIC	Draw Desc	Image
------	-----------	-------

☐ 12. Document ID: US 6119153 A

L3: Entry 12 of 16

File: USPT

Sep 12, 2000

DOCUMENT-IDENTIFIER: US 6119153 A  
TITLE: Accessing content via installable data sources

Detailed Description Text (10):

As shown in FIG. 2, an application 60 (e.g., web browser) of the client system 20 is capable of communicating with a server (e.g., the remote computer 49) in order to download network content 62 therefrom, such as over a dial-up connection. Communication between the client 20 and the server 49 may take place using one of several well-known network protocols, such as hypertext transfer protocol (HTTP), file transfer protocol (FTP), Common Internet File System (CIFS) protocol, or Gopher, although for purposes of simplicity, the invention will be primarily described with respect to HTTP. Content available through these protocols may also be downloaded from the server to the client by alternative means, such as a multicast protocols. As used herein, "server" or "network server" includes any machine or combination of machines having content thereon. Network servers may thus include http "web sites," including those having sites with different names (which may be regarded as different virtual servers even if they are hosted on the same physical machine). Note that a web site may be distributed over many virtual servers, which in turn may be distributed over many physical machines.

Current US Original Classification (1):  
709/218

Current US Cross Reference Classification (2):  
709/225

Current US Cross Reference Classification (3):  
709/232

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWIC	Draw Desc	Image
------	-----------	-------

---

☐ 13. Document ID: US 6108703 A

L3: Entry 13 of 16

File: USPT

Aug 22, 2000

DOCUMENT-IDENTIFIER: US 6108703 A  
**\*\* See image for Certificate of Correction \*\***  
TITLE: Global hosting system

Brief Summary Text (27):

According to the present invention, load balancing across the set of hosting servers is achieved in part through a novel technique for distributing the embedded object requests. In particular, each embedded object URL is preferably modified by prepending a virtual server hostname into the URL. More generally, the virtual server hostname is inserted into the URL. Preferably, the virtual server hostname includes a value (sometimes referred to as a serial number) generated by applying a given hash function to the URL or by encoding given information about the object into the value. This function serves to randomly distribute the embedded objects over a given set of virtual server hostnames. In addition, a given fingerprint value for the embedded object is generated by applying a given hash function to the embedded object itself. This given value serves as a fingerprint that identifies whether the embedded object has been modified. Preferably, the functions used to generate the values (i.e., for the virtual server hostname and the fingerprint) are applied to a given Web page in an off-line process. Thus, when an HTTP request for the page is received, the base HTML document is served by the Web site and some portion of the page's embedded objects are served from the hosting servers near (although not necessarily the closest) to the client machine that initiated the

request.

Detailed Description Text (11):

The routine begins at step 50 by determining whether all of the embedded objects in a given page have been processed. If so, the routine ends. If not, however, the routine gets the next embedded object at step 52. At step 54, a virtual server hostname is prepended into the URL for the given embedded object. The virtual server hostname includes a value (e.g., a number) that is generated, for example, by applying a given hash function to the URL. As is well-known, a hash function takes arbitrary length bit strings as inputs and produces fixed length bit strings (hash values) as outputs. Such functions satisfy two conditions: (1) it is infeasible to

Detailed Description Text (12):

find two different inputs that produce the same hash value, and (2) given an input and its hash value, it is infeasible to find a different input with the same hash value. In step 54, the URL for the embedded object is hashed into a value xx,xxx that is then included in the virtual server hostname. This step randomly distributes the object to a given virtual server hostname.

Detailed Description Text (13):

The present invention is not limited to generating the virtual server hostname by applying a hash function as described above. As an alternative and preferred embodiment, a virtual server hostname is generated as follows. Consider the representative hostname al234.g.akamaitech.net. The 1234 value, sometimes referred to as a serial number, preferably includes information about the object such as its size (big or small), its anticipated popularity, the date on which the object was created, the identity of the Web site, the type of object (e.g., movie or static picture), and perhaps some random bits generated by a given random function. Of course, it is not required that any given serial number encode all of such information or even a significant number of such components. Indeed, in the simplest case, the serial number may be a simple integer. In any event, the information is encoded into a serial number in any convenient manner. Thus, for example, a first bit is used to denote size, a second bit is used to denote popularity, a set of additional bits is used to denote the date, and so forth. As noted above in the hashing example, the serial number is also used for load balancing and for directing certain types of traffic to certain types of servers. Typically, most URLs on the same page have the same serial number to minimize the number of distinguished name (DN) accesses needed per page. This requirement is less important for larger objects.

Detailed Description Text (14):

Thus, according to the present invention, a virtual server hostname is prepended into the URL for a given embedded object, and this hostname includes a value (or serial number) that is generated by applying a given function to the URL or object. That function may be a hash function, an encoding function, or the like.

Current US Original Classification (1):

709/226

Current US Cross Reference Classification (1):

709/105

Current US Cross Reference Classification (2):

709/219

Current US Cross Reference Classification (3):

709/223

Current US Cross Reference Classification (4):

709/224

Current US Cross Reference Classification (5):

709/235

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWAC	Draw Desc	Image
------	-----------	-------

---

☐ 14. Document ID: US 6108701 A

L3: Entry 14 of 16

File: USPT

Aug 22, 2000

DOCUMENT-IDENTIFIER: US 6108701 A

TITLE: Soft switch extension for internet protocol applications

Detailed Description Text (2):

With reference now to the figures and in particular with reference to FIG. 2A, a simplified schematic of the preferred multiple server embodiment of the present invention is depicted. In this embodiment, a plurality of servers 200, 202 are assigned the same IP address on a local area network ("LAN") 204. A router 206 is coupled to the LAN and connects the servers 200, 202 via an Intranet or Internet 208 to a client computer 210. Also coupled to the LAN is a DNS Server 212 that is used for "name to IP address" translation. The plurality of servers 200, 202 thus appear as one logical host (i.e. a virtual server) 214 to the client 210 and also to the DNS Server 212, because the servers 200, 202 share the same IP address and host name.

Current US Original Classification (1):

709/224

Current US Cross Reference Classification (1):

709/203

---

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWAC	Draw Desc	Image
------	-----------	-------

---

☐ 15. Document ID: US 6092178 A

L3: Entry 15 of 16

File: USPT

Jul 18, 2000

DOCUMENT-IDENTIFIER: US 6092178 A

**\*\* See image for Certificate of Correction \*\***

TITLE: System for responding to a resource request

Detailed Description Text (10):

DNS server 100 includes DNS database 102 for resolving client requests. DNS database 102, which may be a zone file or lookup table, thus includes one or more resource records associated with the application (e.g., indexed by a virtual server name or alias by which the application is known or by an identity of a servers hosting an instance of the application). In one embodiment of the invention a new type of resource record is provided which, as described above, references a trigger that is executed upon retrieval of the record.

Current US Cross Reference Classification (1):

709/105

---

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWAC	Draw Desc	Image
------	-----------	-------

---

☐ 16. Document ID: US 6038608 A

L3: Entry 16 of 16

File: USPT

Mar 14, 2000

DOCUMENT-IDENTIFIER: US 6038608 A

**\*\* See image for Certificate of Correction \*\***

TITLE: Virtual LAN system

Brief Summary Text (7):

Further, in order to enable the host terminal to belong to a plurality of virtual LAN's, virtual LAN's to which the host terminal can belong are preliminarily registered in a virtual server for controlling the construction of the virtual LAN's and a connection is made to the virtual LAN server by assigning the virtual LAN to be used in a communication when the host terminal starts the communication.

Brief Summary Text (8):

Since such conventional virtual LAN system determines the upper protocol to be used in a communication every LAN in order to construct a plurality of virtual LAN's every protocol indicative of the communication procedures, a communication within the LAN must be performed by using only this protocol. Therefore, there is a problem that the host terminal can not perform communication by using a plurality of upper protocols within the connected LAN. Further, since, in order to enable the host terminal to belong to a plurality of virtual LAN's, virtual LAN's to which the host terminal can belong are preliminarily registered in a virtual server for controlling the construction of the virtual LAN's and a connection is made to the virtual LAN server by assigning the virtual LAN to be used in a communication when the host terminal starts the communication, there is another problem that can not use other virtual LAN's than the virtual LAN to which the connection is made.

Current US Original Classification (1):709/238Current US Cross Reference Classification (6):709/218Current US Cross Reference Classification (7):709/249

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KnowC	Draw Desc	Image
-------	-----------	-------

[Generate Collection](#)[Print](#)

Term	Documents
VIRTUAL	56040
VIRTUALS	3
SERVER\$	0
SERVER	36940
SERVERA	9
SERVERABILITY	1
SERVERABLE	11
SERVERABLY	2
SERVERACCEPT	1
SERVERACCESS	1
"SERVERACCESS.JAVA"	1
(L1 AND ((VIRTUAL ADJ1 SERVER\$) SAME HOST\$)).USPT.	16

[There are more results than shown above. Click here to view the entire set.](#)

---

**Display Format:**

[Previous Page](#)

[Next Page](#)



Set Name Query

side by side

Hit Count Set Name

result set

*DB=USPT; PLUR=YES; OP=ADJ*L3 L2 and ((private\$ or virtual) and server\$).ab.

8

L3L2 L1 and (tunnel\$).ab.

32

L2L1 ((709/\$)!.CCLS.)

15935

L1

END OF SEARCH HISTORY

## WEST

[Generate Collection](#)[Print](#)

## Search Results - Record(s) 1 through 8 of 8 returned.

☐ 1. Document ID: US 6487598 B1

L3: Entry 1 of 8

File: USPT

Nov 26, 2002

DOCUMENT-IDENTIFIER: US 6487598 B1

TITLE: Virtual dial-up protocol for network communication

Abstract Text (1):

A layer two forwarding protocol (L2F) provides virtual direct dial-up service into private networks through public internet service providers. An authorized remote client appears as a direct dial-up client to the home gateway, even through the client is accessing the home gateway remotely through the ISP. The new forwarding protocol allows the remote client to conduct point-to-point link protocols, such as point-to-point protocol (PPP) and serial line interface protocol (SLIP) directly with the local network home gateway. The network access server changes from a routing mode where a communication protocol is conducted with the client to a switching mode where the POP simply sends data from one port to a tunnel. The tunnel then transmits the data to another port, regardless of the header information on transmitted data packets. The remote client can then be managed through databases controlled by the local network and gain access to resources not typically accessible through the internet. The layer two forwarding protocol conducts an independent authorization session to prevent unauthorized access to the private network and provides point-to-point protocol transport over the internet independently of internet transport protocols.

Current US Original Classification (1):709/227Current US Cross Reference Classification (3):709/201Current US Cross Reference Classification (4):709/229Current US Cross Reference Classification (5):709/230

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KMC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	-----	-----------	-------

☐ 2. Document ID: US 6308213 B1

L3: Entry 2 of 8

File: USPT

Oct 23, 2001

DOCUMENT-IDENTIFIER: US 6308213 B1

TITLE: Virtual dial-up protocol for network communication

Abstract Text (1):

A layer two forwarding protocol (L2F) provides virtual direct dial-up service into private networks through public internet service providers. An authorized remote client appears as a direct dial-up client to the home gateway, even through the client is accessing the home gateway remotely through the ISP. The new forwarding protocol allows the remote client to conduct point-to-point link protocols, such as point-to-point protocol (PPP) and serial line interface protocol (SLIP) directly with the local network home gateway. The network access server changes from a routing mode where a communication protocol is conducted with the client to a switching mode where the POP simply sends data from one port to a tunnel. The tunnel then transmits the data to another port, regardless of the header information on transmitted data packets. The remote client can then be managed through databases controlled by the local network and gain access to resources not typically accessible through the internet. The layer two forwarding protocol conducts an independent authorization session to prevent unauthorized access to the private network and provides point-to-point protocol transport over the internet independently of internet transport protocols.

Current US Original Classification (1):

709/229

Current US Cross Reference Classification (1):

709/201

Current US Cross Reference Classification (2):

709/230

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWAC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	-----------	-------

---

☐ 3. Document ID: US 6269404 B1

L3: Entry 3 of 8

File: USPT

Jul 31, 2001

DOCUMENT-IDENTIFIER: US 6269404 B1

TITLE: Virtual network architecture for connectionless LAN backbone

Abstract Text (1):

Network traffic management is achieved based on automatically setting up a plurality of virtual networks (VNETs) within a single large virtual LAN. Multicast/broadcast traffic is confined to the VNET of the source, without imposing constraints on layer two addressing within the virtual LAN. VNETs are domains of users of a virtual LAN which include members of logical networks defined at layer three or higher. One method includes transferring a multi-destination packet originating from a particular node in the virtual LAN by tunnelling across a connectionless backbone network to a virtual net server. The virtual net server translates the multi-destination packet to a plurality of tunneled messages identifying nodes authorized to receive multi-destination packets from members of the particular VNET which originated the packet. The tunneled messages are then forwarded from the virtual net server to the authorized nodes. This way, multi-destination packets, such as advertisement or discovery packets, are confined to a single VNET. By confining the multi-destination packets to a single VNET, unicast packets generated within the virtual LAN are then also naturally confined to that VNET.

Current US Original Classification (1):

709/238

Current US Cross Reference Classification (1):

709/245

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	-----------	-------

☐ 4. Document ID: US 6101543 A

L3: Entry 4 of 8

File: USPT

Aug 8, 2000

DOCUMENT-IDENTIFIER: US 6101543 A

TITLE: Pseudo network adapter for frame capture, encapsulation and encryption

Abstract Text (1):

A new pseudo network adapter is disclosed providing an interface for capturing packets from a local communications protocol stack for transmission on the virtual private network. The system further includes a Dynamic Host Configuration Protocol (DHCP) server emulator, and an Address Resolution Protocol (ARP) server emulator. The new system indicates to the local communications protocol stack that nodes on a remote private network are reachable through a gateway that is in turn reachable through the pseudo network adapter. The new pseudo network adapter includes a transmit path for processing data packets from the local communications protocol stack for transmission through the pseudo network adapter. The transmit path includes an encryption engine for encrypting the data packets and an encapsulation engine for encapsulating the encrypted data packets into tunnel data frames. The pseudo network adapter passes the tunnel data frames back to the local communications protocol stack for transmission to a physical network adapter on a remote server node. The new pseudo network adapter further includes an interface into a transport layer of the local communications protocol stack for capturing received data packets from the remote server node, and a receive path for processing received data packets captured from the transport layer of the local communications protocol stack. The receive path includes a decapsulation engine, and a decryption engine, and passes the decrypted, decapsulated data packets back to the local communications protocol stack for delivery to a user.

Current US Original Classification (1):709/229Current US Cross Reference Classification (1):709/225Current US Cross Reference Classification (2):709/226Current US Cross Reference Classification (3):709/227Current US Cross Reference Classification (4):709/228Current US Cross Reference Classification (5):709/236Current US Cross Reference Classification (6):709/238Current US Cross Reference Classification (7):709/239

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWIC	Draw Desc	Image
------	-----------	-------

☐ 5. Document ID: US 6081900 A

L3: Entry 5 of 8

File: USPT

Jun 27, 2000

DOCUMENT-IDENTIFIER: US 6081900 A  
TITLE: Secure intranet access

Abstract Text (2):

may be redirected from a target server to a border server, after which a secure sockets layer connection between the border server and the external client carries user authentication information. After the user is authenticated to the network, requests may be redirected back to the original target server. Web pages sent from the target server to the external client are scanned for non-secure URLs such as those containing "http://" and modified to make them secure. The target server and the border server utilize various combinations of secure and non-secure caches. Although tunneling may be used, the extensive configuration management burdens imposed by virtual private networks are not required.

Current US Cross Reference Classification (2):709/230Current US Cross Reference Classification (3):709/245

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWIC	Draw Desc	Image
------	-----------	-------

☐ 6. Document ID: US 6061797 A

L3: Entry 6 of 8

File: USPT

May 9, 2000

DOCUMENT-IDENTIFIER: US 6061797 A  
TITLE: Outside access to computer resources through a firewall

Abstract Text (1):

A firewall isolates computer and network resources inside the firewall from networks, computers and computer applications outside the firewall. Typically, the inside resources could be privately owned databases and local area networks (LAN's), and outside objects could include individuals and computer applications operating through public communication networks such as the Internet. Usually, a firewall allows for an inside user or object to originate connection to an outside object or network, but does not allow for connections to be generated in the reverse direction; i.e. from outside in. The disclosed invention provides a special "tunneling" mechanism, operating on both sides of a firewall, for establishing such "outside in" connections when they are requested by certain "trusted" individuals or objects or applications outside the firewall. The intent here is to minimize the resources required for establishing "tunneled" connections (connections through the firewall that are effectively requested from outside), while also minimizing the security risk involved in permitting such connections to be made at all. The mechanism includes special tunneling applications, running on interface servers inside and outside the firewall, and a special table of "trusted sockets" created and maintained by the inside tunneling application. Entries in the trusted sockets table define objects inside the firewall consisting of special inside ports, a telecommunication protocol to be used at each port, and a host object associated with each port. Each entry is "trusted" in the sense that it is supposedly known only by individuals authorized to have "tunneling" access through the firewall from outside.

Current US Cross Reference Classification (1):709/229

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWNC	Draw Desc	Image
------	-----------	-------

☐ 7. Document ID: US 6041166 A

L3: Entry 7 of 8

File: USPT

Mar 21, 2000

DOCUMENT-IDENTIFIER: US 6041166 A

TITLE: Virtual network architecture for connectionless LAN backbone

Abstract Text (1):

Network traffic management is achieved based on automatically setting up a plurality of virtual networks (VNETs) within a single large virtual LAN. Multicast/broadcast traffic is confined to the VNET of the source, without imposing constraints on layer two addressing within the virtual LAN. VNETs are domains of users of a virtual LAN which include members of logical networks defined at layer three or higher. One method includes transferring a multi-destination packet originating from a particular node in the virtual LAN by tunnelling across a connectionless backbone network to a virtual net server. The virtual net server translates the multi-destination packet to a plurality of tunneled messages identifying nodes authorized to receive multi-destination packets from members of the particular VNET which originated the packet. The tunneled messages are then forwarded from the virtual net server to the authorized nodes.

Abstract Text (2):

This way, multi-destination packets, such as advertisement or discovery packets, are confined to a single VNET. By confining the multi-destination packets to a single VNET, unicast packets generated within the virtual LAN are then also naturally confined to that VNET.

Current US Original Classification (1):709/238Current US Cross Reference Classification (1):709/220Current US Cross Reference Classification (2):709/221Current US Cross Reference Classification (3):709/230

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWNC	Draw Desc	Image
------	-----------	-------

☐ 8. Document ID: US 5918019 A

L3: Entry 8 of 8

File: USPT

Jun 29, 1999

DOCUMENT-IDENTIFIER: US 5918019 A

**\*\* See image for Certificate of Correction \*\***

TITLE: Virtual dial-up protocol for network communication

Abstract Text (1):

A layer two forwarding protocol (L2F) provides virtual direct dial-up service into private networks through public internet service providers. An authorized remote client appears as a direct dial-up client to the home gateway, even through the client is accessing the home gateway remotely through the ISP. The new forwarding protocol allows the remote client to conduct point-to-point link protocols, such as point-to-point protocol (PPP) and serial line interface protocol (SLIP) directly with the local network home gateway. The network access server changes from a routing mode where a communication protocol is conducted with the client to a switching mode where the POP simply sends data from one port to a tunnel. The tunnel then transmits the data to another port, regardless of the header information on transmitted data packets. The remote client can then be managed through databases controlled by the local network and gain access to resources not typically accessible through the internet. The layer two forwarding protocol conducts an independent authorization session to prevent unauthorized access to the private network and provides point-to-point protocol transport over the internet independently of internet transport protocols.

Current US Original Classification (1):709/227Current US Cross Reference Classification (1):709/230

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWIC	Draw Desc	Image
------	-----------	-------

[Generate Collection](#)[Print](#)

Term	Documents
VIRTUAL.USPT.	55214
VIRTUALS.USPT.	3
PRIVATE\$	0
PRIVATE.USPT.	31172
PRIVATEADATA.USPT.	3
PRIVATEATTRIBUTES.USPT.	1
PRIVATEATTRIBUTES-A.USPT.	1
PRIVATEAUDIENCE.USPT.	2
PRIVATEBRANCH.USPT.	1
PRIVATECIRCUITFEATUREAGREEMENT.USPT.	1
PRIVATECLASSES.USPT.	2
(L2 AND ((PRIVATE\$ OR VIRTUAL) AND SERVER\$).AB.).USPT.	8

There are more results than shown above. [Click here to view the entire set.](#)

**Display Format:**[KWIC](#)[Change Format](#)[Previous Page](#)[Next Page](#)